

REMARKS

Claims 1-10 have been rejected under 35 U.S.C. §102(a,e) as anticipated by Sweeney et al (U.S. Patent No. 6,230,314 B1). By the foregoing amendment, Claims 2-7, 9 and 10 have been canceled, Claim 1 has been amended, and Claim 8 has been amended to depend from Claim 1. For the reasons set forth hereinafter, Applicants respectfully submit that amended Claim 1 distinguishes over the Sweeney et al reference. A Request for Continued Examination has been submitted concurrently herewith. Accordingly, the foregoing amendment is entitled to entry and consideration.

The present invention is directed to an automatic software generating system, for automatically generating software from an object-oriented specification. By means of the features of the invention as recited in Claim 1, the software generating system according to the invention results in an improved code efficiency, thereby achieving an economy of time and space.

The software generation system according to the invention includes an optimized information inputting means (described in the specification at page 11, lines 19-25) that permits the inputting, via an external unit, of optimized information (selected information in the set up option 503, as illustrated in Figure 5) indicating the necessity or the lack of necessity for a dynamic generation function for an instance representing one of object-oriented functions; and in addition, the function removing means generates program information in

which an unnecessary function, among a plurality of object-oriented functions, is removed, based on the analyzed specification information, the inputted optimized information and a predetermined function removal rule (function selection item, as shown in Figure 5). Accordingly, through an input of optimized information from an external unit (such as a keyboard), and through generating program information in which an unnecessary function is removed based on the input information, the use/disuse selection of the dynamic generation function in the instances can be performed depending on the devices executing the program. As a result, the size the program is reduced as a whole, and the memory capacity necessary in order to store the program is reduced as well.

In addition, the above "optimized information" implies information in which the use/disuse of the dynamic generation function in the instances is selected. With the "optimized information", a user (that is, an operator of the system) inputs "not to use" for the dynamic generation function in instances where there is a single object (with regard to for example, a sensor in a monitor mounted or connected to the device which executes the program), and inputs the "use" in cases where there are plurality of objects. In this manner, a further reduced program size can be obtained, and the memory capacity can be further limited.

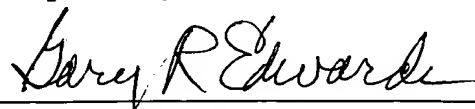
Applicants respectfully submit that Claim 1 as amended distinguishes over Sweeney et al. In particular, Sweeney et al nowhere discloses the feature of the present invention of providing the optimized information inputting means, which permits an operator to input, via an external unit, optimized information

indicating a need or lack of need for dynamic generation function of an instance representing one of object-oriented functions. While Sweeney et al discloses the analysis of information necessary for optimization from programs, with such a measure, it is impossible to set a dynamic generation function in instances which do not depend on programs. As a result, because of the inclusion of useless program information, the program size reduction is not achieved, and the reduction of memory capacity is also limited.

In light of the foregoing remarks, this application should be in condition for allowance, and early passage of this case to issue is respectfully requested. If there are any questions regarding this amendment or the application in general, a telephone call to the undersigned would be appreciated since this should expedite the prosecution of the application for all concerned.

If necessary to effect a timely response, this paper should be considered as a petition for an Extension of Time sufficient to effect a timely response, and please charge any deficiency in fees or credit any overpayments to Deposit Account No. 05-1323 (Docket #381NP/47598).

Respectfully submitted,

A handwritten signature in cursive script, reading "Gary R. Edwards", is written over a horizontal line.

Gary R. Edwards

Registration No. 31,824

CROWELL & MORING, LLP
Intellectual Property Group
P.O. Box 14300
Washington, DC 20044-4300
Telephone No.: (202) 624-2500
Facsimile No.: (202) 628-8844
GRE:kms
(CAM 56207.324)

VERSION WITH MARKINGS TO SHOW CHANGES MADE

Please amend Claims 1 and 8 as follows:

1. (Twice Amended) A software generation system comprising:

a specification analysis means which analyzes an object-oriented specification for deriving specification information;

an optimized information input means for inputting optimized information from an external unit, indicating a necessity or lack of necessity for use of a dynamic generation function for dynamic generation of an instance representing one of a set of object oriented functions;

a function removing means which checks said specification information derived by said specification analysis means and the optimized information input via said optimized information inputting means by collating with a predetermined function removal rule, which [is predetermined, and] removes a function which becomes unnecessary from a set of object-oriented functions by which members are realized, for generating from the specification information and the optimized information, program information excluding the unnecessary function; and

a code generation means for generating a code according to said program information obtained by said function removing means.

8. (Twice Amended) [A] The software generation system according to Claim 1, further comprising:

[a specification analysis means which analyzes an object-oriented specification for deriving specification information;]

an analysis result display means for displaying a status of use of an object-oriented function by which a member is realized from [said] the specification information. [;]

[an input means whereby to select an object-oriented function to utilize;]

[a function memory means for storing a function selected via said input means;]

[a program information generation means for generating program information on the basis of said specification information derived by said specification analysis means and using said function selected and stored in said memory means; and]

[a code generation means for generating a code on the basis of said program information obtained by said program generation means.]